Poseidon House
Castle Park
Cambridge CB3 0RD
United Kingdom

TELEPHONE: **Cambridge (01223) 515010**
INTERNATIONAL: **+44 1223 515010**
FAX: **+44 1223 359779**
E-MAIL: **apm@ansa.co.uk**

ANSA Phase III

Reflective Java (Slides for Marketing)

Zhixue Wu

Abstract

This is a simple presentation about Reflective Java for marketing purpose.

APM.1972.00.01 **Draft** 13th March 1997

Marketing and Contracts

Distribution:
Supersedes:
Superseded by:

Reflective Java:
A Powerful Tool to Produce Flexible and Adaptable System Software

APM Ltd.

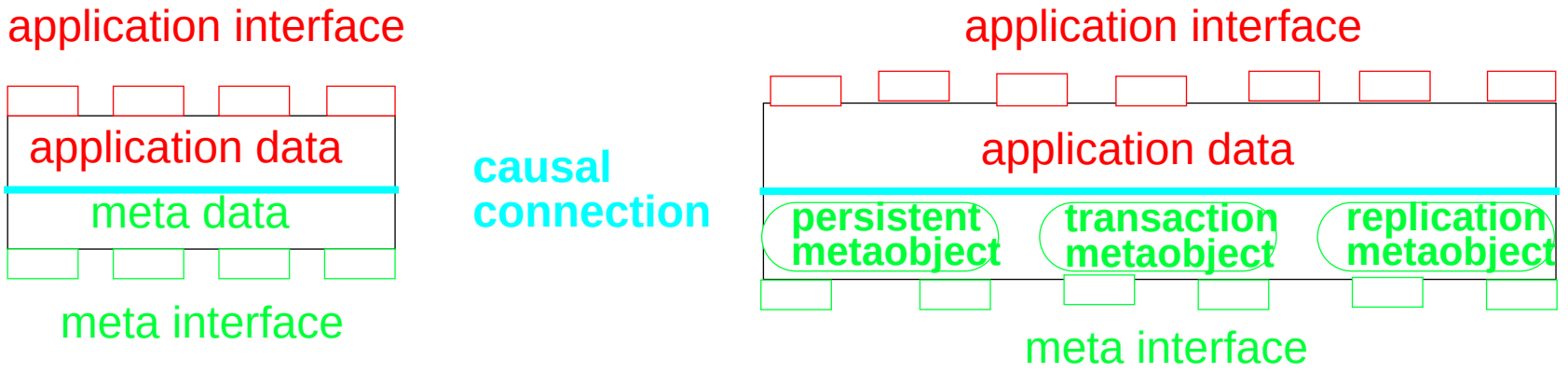
1997

Problems

- The one size fits all design strategy becomes obsolete
  - mew applications: mobile computing, internet programming, and multimedia applications
  - different considerations and requirements for underlying system software
- The black box approach becomes inappropriate
  - different configuration
  - different quality of service
- System must be made flexible and customisable at run time
  - many attributes of the application environment vary from time to time, and from place to place

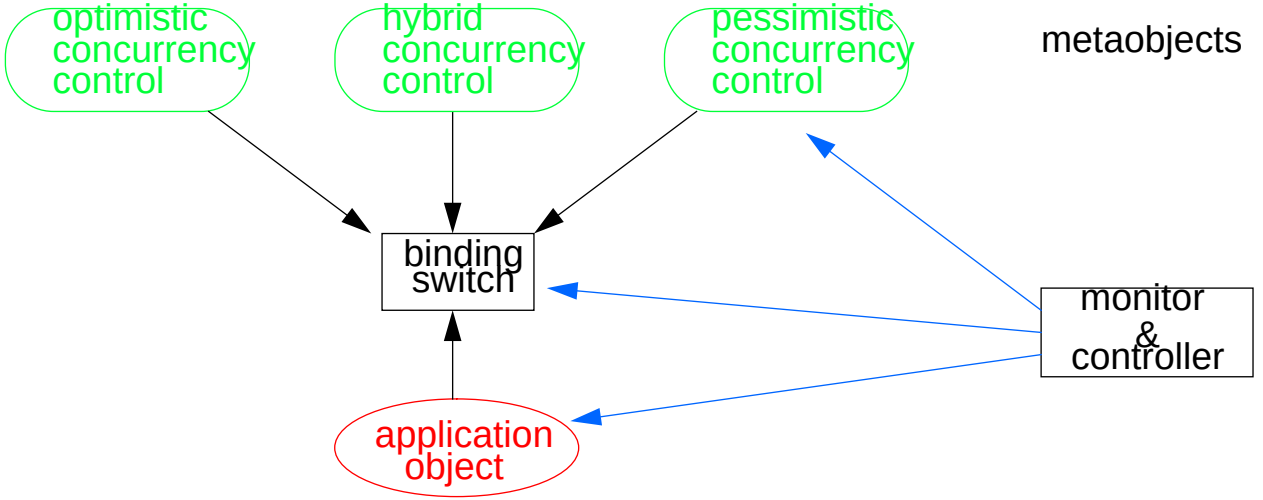
Solutionn: Make System Reflective

- **Reflection**
  - the capability of a system to reason about and act upon itself and adjust itself to changing conditions
  - opens up a system's implementation in a principled way
  - provides an abstraction of the system's behaviour and internal state at the meta level
- **Metaobject protocol = reflection + object-oriented programming**
  - represents the system at the meta level using a family of meta objects
  - allows the system's behaviour to be locally and incrementally adjusted



Idea

- Make a clear separation between functional and non-functional components
- Functional requirements are satisfied by application objects
- Non-functional requirements are satisfied by metaobjects
- Non-functional capabilities are added to an application object by binding it to an appropriate metaobject
- Customisation can be done by switching a binding between an application object and its metaobject



Reflective Java

- Enable Java-powered system to be customised according to the particular requirements of applications and run-time environment
  - statically at compile time
  - dynamically at run-time
  - flexibly
  - transparently
- Make Java reflective
  - without any change to the language itself
  - without any change to its compiler
- without any change to its virtual machine

Benefits

- Easy to upgrade products in order to adapt to changes: either in hardware, in software, or in application requirements
- Flexibility to customise policies dynamically to suit run-time environment
- High-level transparency to applications
- Write an application once, run it anytime, anywhere, in any environment, with any “-ability”
- Flexible configuration
- Free choice of components